Understanding Responsive Web Design

A detailed look at the current state of mobile web site development options.
“Most people make the mistake of thinking design is what it looks like. People think it’s this veneer – that the designers are handed this box and told, ‘Make it look good!’ That’s not what we think design is. It’s not just what it looks like and feels like. Design is how it works.”

– Steve Jobs

Summary

The consensus today is that organizations need a mobile web marketing presence to appear credible. But how does one provide the best mobile web experience that also functions as an integral part of overall online customer experience?

One of the most common – yet widely misunderstood – solutions is the Responsive Web Design approach. This white paper discusses Responsive Web Design patterns and frameworks, the pros and cons of Responsive Web Design (RWD), alternatives to RWD and how to create your mobile web strategy.
Do I Really Need a Mobile Web Presence?

Here are just a few headlines from blog posts and news in 2013:

- “There Will Be More Smartphones Than Humans on the Planet by Year’s End”
- “Mobile to overtake fixed Internet access by 2014”
- Web Traffic From Mobile Devices Up 78% Year Over Year
- 56% of American Adults are now Smartphone Owners

The chart below shows the monthly trend of mobile Internet browsing statistics published by Stat Counter. Most projections now predict that by mid 2014, 50% of web traffic will be from mobile devices.

The Federal Trade Commission (FTC) has made a mobile optimized website part of their core requirements. The Canadian Government published Standards on Optimizing Websites and Applications for Mobile Devices in April 2013.

One increasingly common option for building mobile friendly web sites is the Responsive Web Design approach. At the end of 2012 Mashable posted a story: “Why 2013 Is the Year of Responsive Web Design” and in March 2013 Forbes’ web site published an article by Susan Gunelius: “Why You Need to Prioritize Responsive Design Right Now”. But while clients have woken up to the concept and commonly request a responsive website redesign, it is usually without sufficient understanding of the efforts or process involved.
What is Responsive Web Design?

The main idea behind responsive web design is that the site layout changes dynamically to adjust to the users screen resolution and orientation.

The term Responsive Web Design is relatively new, having been first introduced by Ethan Marcotte on May 25, 2010 in his article entitled “Responsive Web Design.”

Google Search Trends illustrate just how recent this terminology is, and how rapid its adoption has been.
Why We Need New Approaches to Support Mobile Users

Modern smartphones have Retina displays with a very high screen resolution allowing them to display full desktop versions of a site. For example, the iPhone 5 has a 1136 X 640 display that is comparable with a default 1024X768 desktop computer’s resolution. Many web sites simply ignore the difference and hope that users with modern phones will access their website. Below is an example of how a regular desktop web site whose URL was accessed from a printed QR code is rendered on an iPhone 5 with a retina display.

As we see from this example, there are two major issues when displaying a regular website on a high-resolution mobile screen: Readability and Touch Friendliness.

Readability research shows that a font size of 3.5 mm (10 printer points) is sufficient for the 20-49 age group, while older viewers require a font size of at least 4.2 mm (12 points). To provide a readable font size we need many more pixels on a retina display for one character than on a desktop, and twice as many as the iPhone’s previous generation. To make the translation to retina devices easier, Apple introduced a virtual (logical) screen to scale content properly from virtual to physical pixels on retina devices. Apple also introduced a new virtual screen size unit – “point” to decouple physical pixels from virtual screen units.

Touch Friendliness deals with the size of elements that support gesture-based navigation and form entry with a slide-in virtual keyboard. The key principle is that we require enough space between clickable elements to make touch and gesture navigation easy. The hit area should be at least 44 x 44 iOS points or about 9mm². Many typical desktop web sites have navigation elements and menus that are not friendly for small multi-touch screens. So, if a web site has input forms even as simple as in the example above, it requires a completely different design to make it usable on mobile devices.
The Main Elements of the Responsive Web Design

Media Queries

Web developers use media queries in order to match style sheets to a particular screen resolution and form factor. Media Queries allow the system to determine device Media features, such as:

- Page Width
- Page Height
- Device-width
- Device-height
- Orientation
- Aspect-ratio
- Device-aspect-ratio
- Color
- Color-index
- Monochrome
- Resolution
- Scan
- Grid

The World Wide Web Consortium (W3C) introduced media queries as a Candidate Recommendation on 8 July 2002 as part of HTML 4 and CSS2. W3C made media queries an official recommendation in June 2012.

Breakpoints

Breakpoints use media queries with predefined values that specify when to change the page layout to match the screen size and aspect ratio. Because of the differences in pixel density (number of pixels per inch) for different screens, we cannot rely on a simple device pixel width to define breakpoints. This is because a user’s distance from the screen – and therefore the size of individual elements – will also need to adjust between larger desktop screens, tablets and smartphones. So, how do we define when the site layout should break? The best approach is to adapt breakpoints using media queries in relative typographic EM units that return results based on the browser’s default font size.

Fluid Proportional Grids

Fluid proportional grids form the underlying structure for the page content, with sizing based on percentages vs. pixels (or points), that allow the system to scale the design within certain limits or until it reaches a break point.

Flexible Images

Flexible Images also are able to scale their size in relative units using a CSS tag. Media Specific Cascading Style Sheets (CSS) use standard CSS3 that supports CSS selection based on media queries.
Responsive Design Patterns

Luke Wroblewski provided an overview of the most popular Responsive Web Design patterns.

Mostly Fluid

The most popular pattern of the first generation of responsive design, the Mostly Fluid pattern uses a percentage width for screen elements. Content containers are flexible and allow the layout to stretch or contract up to a predefined break point, and then will stack columns once that breakpoint is reached. The Mostly Fluid pattern requires a limited number of breakpoints. It also defines the maximum width that the layout’s whitespace may expand beyond the boundaries after this maximum is reached.

Column Drop

Unlike the Mostly Fluid design pattern, the Column Drop pattern primarily uses fixed sized grid columns to support fixed sized content containers. When the browser width does not fit the layout, columns drop below each other. The Column Drop pattern therefore requires more break points to stack column containers.
**Layout Shifter**

The *Layout Shifter* pattern changes layouts completely to adapt to various screen dimensions. This pattern typically provides a better user experience and control over design than the *Mostly Fluid* and *Column Drop* patterns. This approach is very flexible and allows for innovative patterns, but also requires more time and effort to develop.

**Tiny Tweaks**

This pattern works for simple one-column layouts, when a layout may be adjusted by just tweaking font and graphics sizes. It is not applicable in most cases since the majority of web sites have more complex layouts.

**Off Canvas**

The *Off Canvas* pattern uses space off the screen to hide content or navigational elements that will become visible on larger screen sizes or when the user exposes it on a smaller device. It is a popular pattern that has multiple variations in implementation. It is also a scalable pattern that can be used for complex multilevel navigation with animated sliding menus. It does, however, requires the use of JavaScript/jQuery animation that can affect performance on mobile devices.
Responsive Web Design Frameworks

Mobile First vs. the First Generation of Responsive Frameworks

The still popular first generation of responsive frameworks uses media queries to detect the device's browser screen resolution and follows breakpoint rules to change layouts. The default is a three column desktop layout that can be contracted until it reaches the defined breakpoints wherein the layout changes from three columns to two and then from two to one column, stacking the content blocks using one of the patterns described above.

By contrast, a new generation of responsive web design frameworks such as Twitter Bootstrap V.3 and Zurb Foundation V.4 are based on the Mobile First approach. With a Mobile First approach, the starting point is a smartphone screen, with design and content elements added sparingly for tablet, medium desktop or large screen desktop. The idea behind this approach is to deliver an optimal mobile experience instead of a compromised version of the desktop view. Proponents of mobile first argue that this in turn provides an optimized experience for large screen devices by limiting the amount of information presented in each view, rather than filling every pixel on the screen with additional content choices simply because there is space available.

Mobile Last
Graceful degradation of user experience

Mobile First
Progressive enhancement of user experience
Twitter Bootstrap is the most popular responsive framework with the largest user base. The first release of Twitter Bootstrap arrived in August 2011. Bootstrap 3 was released in August 2013 and was completely redesigned to follow the Mobile First principles described above.

Bootstrap 3 uses one grid to support fluid and fix layouts. It also supports nested containers within the grid. There are 4 defined breakpoints:

- phone < 768 px,
- tablet > 768 px,
- medium desktop > 992 px and
- large desktop > 1200px.

The framework partially supports HTML5 and CSS3 and it is compatible with all major browsers including Microsoft IE8. Twitter Bootstrap is a modular framework that includes a series of LESS (Dynamic Stylesheet Language) style sheets. Twitter Bootstrap includes an extensive library of UI components for buttons, groups of buttons, horizontal and vertical tabs, navigation, breadcrumb navigation, pagination, progress bars and more. The compressed size of the framework is 83.9 KB (archived), making it suitably lightweight for mobile devices.

Foundation

Zurb released the Foundation 4.x framework at the beginning of 2013. Like Twitter Bootstrap 3, The Foundation version 4.x has been rewritten using the Mobile First paradigm. The Zurb Foundation framework has a smaller user base than Twitter Bootstrap but the popularity of this framework has been growing since the release of Foundation 4.

Foundation 4 uses fluid grids with a customizable number of columns from 1-16 and default grid with 12 columns. The Foundation framework includes HTML and CSS–based design templates, an extensive library of user interface components and JavaScript extensions.

It uses a single breakpoint at 768 px and media queries for visibility classes that allow it to alter styles for screens at different sizes and orientation. The Foundation partially supports HTML5 and CSS3 and it is compatible with major browsers – though it does not support Microsoft IE8. The download–archived size is 149.6 KB.
Other Mobile Web Frameworks

**Skeleton**

The **Skeleton** framework is a lightweight framework that is based on a 960 pixel 16 column grid and scales down for mobile screens. It is a “mobile friendly” rather than **Mobile First** framework. **Skeleton** does not provide an extensive UI library and has a much smaller user base than **Twitter Bootstrap** and **Foundation**.

**HTML 5 Boilerplate**

**HTML 5 Boilerplate** is the first and the most popular library for the development of web sites and web applications using HTML5. It includes a mobile friendly HTML template, placeholder touch-screen icons, normalize.css, Google analytics mobile snippet and other mobile-focused tools. The normalize.css file allows browsers to render all elements more consistently. It is optimized for fast mobile site performance. **HTML 5 Boilerplate** uses Zepto.js – minimalist Java script library for modern browsers.

**HTML 5 KickStart**

**HTML 5 KickStart** is a lean library of HTML5, CSS and JavaScript building blocks for rapid website production. It includes optional responsive flexible grid, set of UI components and extensive library of 249 icons.
Responsive Web Design for eCommerce Sites

Responsive design faces many more challenges with eCommerce sites and other transactional web applications than with standard informational sites.

Recent statistics show very high bounce and drop off rates during the mobile checkout process as users are challenged with entering information on mobile devices using small touch screens with intrusive virtual keyboards. Many of the standard patterns found on eCommerce sites do not transfer easily to smaller touch screens. In many cases organizations have to develop a completely separate mobile eCommerce site that shares backend functionality or implement a mobile optimized UI level. While relatively simple eCommerce sites can be handled with responsive frameworks, complex eCommerce sites cannot practically be re-developed using common responsive web design techniques. The exponential complexity of providing different navigation patterns and views on different screen sizes and orientations makes a separate mobile experience more practical.

Another key factor is speed. Responsive sites may be slower to load and render than mobile optimized non-responsive sites. Users today have high expectations around eCommerce site performance, which in turn can have a direct impact on conversion rate and sales.
Implementing Responsive Design with CMS

There is often confusion about whether an existing web content management system (CMS) can be used with a responsive design. Historically most of the major web CMS platforms do not have user-friendly tools for mapping content elements to different device requirements in order to build responsive, multi-device web sites.

To overcome the limitations of a non-mobile web CMS, developers have created responsive themes, of which there are now many available. Let’s review some of popular CMS platforms and their support for Responsive Web Design.

WordPress

Pros

WordPress is a blog publishing platform that has evolved to become the most popular web content management systems (CMS) for personal and simple sites, now powering over 70 million websites. The huge WordPress user base has given birth to an ecosystem of plugins and themes that allow one to extend the platform to create highly specialized sites with multiple features, including responsive design themes.

Cons

Setting up a web site with a responsive web design theme without customization should, in theory, be a relatively simple exercise. However any customization required - and there is almost always some customization desired - can quickly make the process significantly more challenging. Even simple customization can break the integrity of the theme and produce unpredictable results that may not be immediately apparent. Therefore, it requires advanced CSS and WordPress skills to convert even a moderately complex WordPress site to a responsive site.

In addition, speed and performance are likely to be an issue. Many responsive WordPress themes have no server side image optimization and require mobile devices to download the same large images as non-mobile devices and then scale them down to fit the display, rendering the site slow and falling short of an acceptable user experience.
**Drupal**

Pros

Drupal is one of the top three open source Content Management Systems together with WordPress, and Joomla. The Drupal community offers a number of modules that help to build responsive web sites. The AdaptiveTheme framework includes a pluggable layout system, SASS integration and support for Mobile First and desktop first design methods. Other powerful Drupal modules are Adaptive Image, Retina and Picture, which optimize image sizes and resolution for different devices, allowing pages to load and render much faster than with CSS fluid image resizing. Drupal also has responsive modules like FidVids to handle video on mobile devices.

Cons

The core Drupal architecture was not originally designed as a responsive CMS. Nevertheless, Drupal’s modular approach provides enough flexibility to deliver complex responsive web sites.

**Adobe® Experience Manager Web Content Management (former CQ5)**

Pros

Adobe Experience Manager is an open, standards-based web content management platform for delivering multi-channel experience. With full separation of content and presentation logic, it enables marketers and developers to provide an optimal, personalized experience on different devices. This is an enterprise level system that supports multi site and multi device delivery, content workflow, and digital asset management. It has a powerful user interface with in-place content editing and drag and drop page assembly.

Adobe Experience Manager can also be integrated with the Adobe PhoneGap development framework to create mobile web applications. Adobe’s platform is one of the most comprehensive platforms for multi device web design and development and has been named a leader in web content management by Gartner.

Cons

Adobe Experience Manager is designed for enterprise clients, and therefore targets the high end of the CMS spectrum. It is likely not a good fit for small and medium size business due to the high licensing and implementation costs.
Responsive Web Site Analytics

Responsive Web Design requires that we rethink traditional approaches to web analytics. With a responsive site, visitors on different devices, may see a dramatically different presentation of content and navigation links above the fold. Therefore the user’s screen flow will also vary by device (A “screen flow” being the set and sequence of screens A user sees based on his navigation choices). To properly understand user behavior in this context we must first segment site visitors based on device form factor. We can then build a new model for analytics that focuses on understanding the visitor’s interaction with specific content rather than with a page or screen view. This in turn allows us to gain a better understanding of a visitor’s entire journey through the site instead of a single visit to a page.

Web traffic sources and referral pages will also likely differ per device. For instance, marketing campaigns commonly refer traffic sources to a mobile web with a printed QR code. It is a good idea to have landing pages associated with a specific campaign tagged to measure the impact of different QR codes on traffic to your mobile site. You may also need to create distinct tags between mobile and desktop devices using media queries.

The specific tagging approach ultimately depends on your chosen web analytics platform. However, in general responsive sites require you to invest significantly more time in planning and tagging implementation to ensure accurate and meaningful data.

Below are examples of a QR code tag and the landing page that is served when a user scans the printed QR code using the iPhone scan application.
Testing and QA for Responsive Web Site

Because of the complexity of responsive sites, and the rapidly evolving mobile device landscape, it is nearly impossible to ensure compatibility with all device form factors and browsers. Using available browser usage statistics we can make assumptions about what share of the potential audience to target with compatibility testing, and then choose a level of compromise that we deem to be acceptable. The decision will ultimately be driven by the amount of additional effort involved, since moving from a 90% compatibility target to a 99% target may increase the number of devices and browsers to be tested exponentially.

Tools for Testing

There are number of tools that allow us to test sites for different screen resolutions. Device simulators can be used on a desktop computer to test portrait to landscape rotation and other simulated hardware features. However, be warned that device simulators do not always exactly match the native device’s behavior. It is strongly recommended that you perform early testing of a site’s design and development iterations on the most popular devices and browsers. For instance, some of the Java script libraries’ compatibility problems may cause a complete browser failure on certain devices and destroy your entire website user experience. If it is essential that you reach 99.99% of your potential audience, you should consider partnering with a specialized QA service provider who can provide multi-device testing with a high level of automation.

Below are recent statistics about mobile browsers and mobile screen resolutions.

Top 9 Mobile Browsers
From Sept 2012 to Sept 2013

<table>
<thead>
<tr>
<th>Mobile Browsers</th>
<th>Sept 2012</th>
<th>Sept 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>40%</td>
<td>32%</td>
</tr>
<tr>
<td>iPhone</td>
<td>8%</td>
<td>16%</td>
</tr>
<tr>
<td>Opera</td>
<td>24%</td>
<td>32%</td>
</tr>
<tr>
<td>UC Browser</td>
<td>32%</td>
<td>24%</td>
</tr>
<tr>
<td>Nokia</td>
<td>8%</td>
<td>8%</td>
</tr>
<tr>
<td>BlackBerry</td>
<td>4%</td>
<td>2%</td>
</tr>
<tr>
<td>Netfront</td>
<td>8%</td>
<td>8%</td>
</tr>
<tr>
<td>iPod Touch</td>
<td>12%</td>
<td>10%</td>
</tr>
<tr>
<td>Chrome</td>
<td>12%</td>
<td>10%</td>
</tr>
<tr>
<td>Other (dotted)</td>
<td>4%</td>
<td>4%</td>
</tr>
</tbody>
</table>
Performance Testing for Responsive Sites

Users browsing web sites on smartphones have a very low tolerance for slow loading pages. Statistics show that every second of additional page load delay dramatically increases the number of visitors who abandon the site. Web page load and rendering speed is the area where Responsive Web Design is most likely to fail and therefore you should plan to at least double your time investment in performance testing with a responsive site.

There are a number of optimization techniques which can help with the problem of lowered performance. However you will need to begin your on native devices at the earliest possible stages of design and prototyping to avoid a costly reworking at the end of the project.
Pros and Cons of Responsive Web Design

Pros of Responsive Web Design:

• It allows you to build one website instance that works on all devices and simplify the maintenance of web content and application code

• Users do not need to download a stand-alone application to enjoy a mobile optimized experience

• There are advantages for Search Engine Optimization (SEO) since there is no URL redirection required and content is shared across all platforms

Cons of Responsive Web Design:

• Contextual factors such as location, user surroundings, number of steps and time required to complete task, are fundamentally different on hand held devices.

• Multi touch screen interaction requires not just screen size adjustments but a rethinking of the overall navigation architecture.

• We may be forced to settle for the lowest common denominator from a design perspective in order to minimize the risk of having an unpredictable user experience.

• A responsive site demands that Mobile devices perform additional calculations to load, sort, resize and deliver specific content elements and styles for each device. Because mobile users have a very low tolerance for sites with page load time over 3 seconds, this may have serious implications for the success of the site.

• There is a common misperception that developing a responsive site is less expensive than developing separate sites for each device form factor. In reality, the efforts required to design, develop and properly test a custom responsive site can be 2-3X greater than a site that targets only one device.

• Generally, having 3 sites that are optimally designed for Desktop, Tablet and Smartphone provides a better user experience than having a responsive web site that provides a second-rate experience on all devices. The typical responsive design framework considers only a limited number of features and factors pertaining to a user’s device.
Device Features that Impact User Experience Across Different Platforms:

• Physical screen dimensions in pixels
• Aspect ratio
• Rotation between landscape and portrait orientations
• Screen pixel density in Pixels Per Inch (PPI). The retina display for iPhone has 340 PPI while the typical notebook display has 40 – 50 PPI
• The base font size for readability on the physical device
• Screen brightness contrast and ambient light sensor
• Processor and graphical speed
• Wi-Fi or cellular network speed
• Input methods: fixed keyboard, on screen sliding keyboard, mouse or pointer, touch panel, single touch screen, multi-touch
• Additional gestures supported by specific device type
• Microphone and audio support
• Camera + bar code reader support

• GPS and location based services
• Availability of push notifications
• Three-axis gyro
• Accelerometer
• Proximity Sensor
• Biometric security – fingerprint identity sensor
• Near Field Communication (NFC)
• Impact of web applications on smartphone batteries
Responsive Web Design Alternatives

Adaptive Web Design

Adaptive Web Design methodology uses server side processing to deliver a better user experience by determining an optimal compromise between site features and device capabilities. This approach is based on the principle of “Progressive Enhancement,” whereby the site proactively eliminates the need to download unnecessary components by a browser or device that is unable to support them.

Responsive Design + Enhanced Server Side Components (RESS)

Combining Responsive Design and Enhanced Server Side components (RESS) improves the performance of responsive web sites by using server side logic to send the device only the content, optimal styles, images and JavaScript that are required by that specific device type.

The RESS approach utilizes fluid grids to accommodate different screen dimensions using the responsive design approach that is rendered by the device browser.

This saves time and bandwidth, as well as processing time to execute scripts, or render styles and images on mobile screens.

Here is a summary of the key differences between Responsive Web Design (RWD), Adaptive Web Design (AWD) and Responsive Design + Enhanced Server Side components (RESS):

<table>
<thead>
<tr>
<th>Type of Responsive Web Method</th>
<th>Where the Responsive Web Logic is Handled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsive Web Design (RWD)</td>
<td>On the device browser</td>
</tr>
<tr>
<td>Adaptive Web Design (AWD)</td>
<td>On the server side</td>
</tr>
<tr>
<td>Components (RESS)</td>
<td>Device</td>
</tr>
</tbody>
</table>
Mobile Responsive Delivery

Moovweb introduced the concept of Responsive Delivery, which eliminates the need to rewrite an existing desktop-based web site to provide an optimal mobile device experience. Responsive delivery adds a virtual presentation layer (t.engine) that transforms an existing site to a virtual site that matches a device’s capabilities in real time. This provides a rapid development process for mobile web sites, since you only need to design the mobile user interface and code style transformation rules for different device types.

The main advantage of Moovweb’s Responsive Delivery methodology is that it allows you to leverage your investment in an existing desktop web site or web based application while still providing an optimal user experience on each device.
Native Mobile Applications

Native mobile applications (i.e. non-browser based) have the potential to provide better performance and user experience since they integrate with advanced device hardware features, and include the ability to provide offline functionality. Because they are developed for each particular device operating system, they require users to download an “app” from one of the OEM’s online marketplaces – Apple App Store, Google Play, BlackBerry World or Windows Store – which is a fairly high barrier for a one time or occasional user of your site.

The main challenge with native mobile applications is the necessity to develop and maintain separate applications for each device. While there are tools and platforms that help to simplify cross platform mobile development, this solution is generally only feasible where frequent, dedicated users will take the time to download an additional application.

Web-based Mobile Applications

HTML5 now allows us to develop web-based applications that are very close to the look and feel of native mobile applications – but which can be delivered within a mobile web browser. With Mobile HTML5 applications, mobile browsers gain direct access to the hardware of mobile devices (including accelerometers and GPS chips) allowing for a very sophisticated experience, including caching of user data, without the necessity of downloading a dedicated app or maintaining versions for multiple devices. While traditionally limited to a subset of mobile devices, HTML5 is now available in nearly all modern mobile browsers.

Hybrid Mobile Applications

Hybrid mobile applications combine modern web technologies such as HTML5 and CSS3 while utilizing essentially the same device features as native mobile applications. The idea behind hybrid web application frameworks is to deliver the best possible web mobile experience using the Web View Shell. The Web View Shell is a native mobile application that works as a container for the core application HTML5 code. We need to develop and distribute the Web View Shell applications for each device in the same way we develop and distribute native mobile applications.

There are two types of hybrid applications.

- The first type uses only native shells but all application code is built as a cross platform web application. You only have to submit native mobile applications shells to the various application stores to cover every device. Developers do not need to develop the core application code for each device.
The second type of hybrid application mixes the idea of a mobile web application with native mobile application code modules that helps to enhance the user experience. The second type is closer to native mobile applications, but a significant part of the core application code can be reused between devices. This approach also has an advantage with cross device updates because there is only one core application HTML5 code distribution required.

Criteria for Selecting a Mobile Web Strategy

The first step is to assess your current web properties and mobile priorities. Here are a few questions that can help you make an optimal choice that addresses both user needs and business objectives:

• What type of site is it?
  • A small corporate site (up to 10 page template layouts)
  • An eCommerce site
  • Online publishing site
  • Portal

• Complex multifunction site
• Multi-site framework or network of multiple sites

• How old is your current web site? Has there been a complete refresh of the website in the last 1-3 years or it is older?

• Who are the target audiences for your sites and what web browsing statistics are expected? This is particularly important if you have a site with global audience.

• Does your current site use Adobe Flash components or flash based video? (these are not supported by many mobile devices)
A Summary of Best Practices for Implementing a Responsive Website

Our recommendation is that a responsive approach is most appropriate for small corporate or eCommerce sites (i.e. no more than 10 distinct page layouts). If you decide to move forward with a Responsive Web Design approach here is a summary of best practices that will increase your chances of a successful implementation:

- Follow the mobile first approach described above. It will provide a better overall user experience.
- Start with a full list of content and user actions, then prioritize each of them by device.
- Prepare user experience flows for each device.
- Select navigation menu patterns per device that are multi-touch friendly and match the hierarchical structure of your site. You will likely need to apply a different menu hierarchy on mobile phones vs. a larger screen. You shouldn’t have any more than 3 navigation items in a horizontal menu on a smartphone device screen.
- 44 x 44 points is the minimum size for touch based clickable elements.
- You will need to rethink pointer based hover functionality for touch screens.
- Use adaptive media technologies for images and video. Avoid loading and resizing large images using CSS on smartphones.
- Use optimized JavaScript libraries and adjust the priority of script preloaded to improve the page rendering speed on mobile devices.
- Start testing device compatibility and page speed on the primary native devices you are targeting early in the process to avoid a major rework at the end of your project.
- Plan your web analytics implementation in parallel with the site map and experience flow phases to insure you are collecting relevant data.
- Use mobile web optimized tagging.
- Expect to allocate 2–3X more effort and duration in comparison with a similar sized single device web site project for the entire cycle, including planning, design, development and testing.
If you have complex functional or transaction based sites then the Responsive Delivery framework will likely prove more effective and save you a lot of time and effort in converting or adapting a complex web site or application to a multi screen experience. Responsive delivery platforms also help simplify and unify the transformation of multiple web sites for mobile use and preserve your investment in the development of existing complex functional web sites.

Conclusion

Having a mobile web site is no longer simply an option – it is a necessity. And choosing the right mobile approach is a critical factor in maximizing your online marketing and business results.

Using a Mobile First design philosophy is likely to deliver the best results regardless of which mobile web approach or platform you choose. However, as with any strategy, you should base your multi screen approach on a thorough understanding of your audience’s needs and your business objectives.
References

1. There Will Be More Smartphones Than Humans on the Planet by Year’s End http://mashable.com/2012/02/14/more-smartphones-than-humans/ 
21. Drupal, https://drupal.org/node/1388492

Read more: http://www.marketingprofs.com/charts/2013/11010/web-traffic-from-mobile-devices-up-78-year-over-year#ixzz2hmrFC700